iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 Issue: III Month of publication: March 2018

DOI: http://doi.org/10.22214/ijraset.2018.3605

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Traffic Analysis, Monitoring and Control using Agent Technologies (Simulation)

Yash Kesarkar[1], Siddhi Nikam[2], Vaishnav Khandekar[3], Prajkta Akre[4]
[1, 2, 3, 4,] Department of Computer Engineering, St. John College of Engineering and Management, University Of Mumbai

***Abstract-:* Traffic monitoring is essentially one of the most sought problem of civil administration in recent times. More-so ever the problem of traffic resolving is becoming grave every passing day this can be detect by using a tracking beacon methodology mechanism which can keep a track of which is in real time. The proposed system aims to build this mechanism put into test with the most of extreme physical conditions on the road which could be emulated the effect and also resolve traffic on a timely basis. The system will be build using randomization techniques for traffic. The optimization of road traffic is researched mostly on remote sensing but in a case to case scenario, remote sensing applications without any fixed pivot technologies like Global Positioning System (GPS) can provide very dated data which is of little help in real time traffic resolution. Remote Sensing comes handy when traffic is to be mapped on a freeway or a expressway as the motion of objects being sensed is high. On the other hand, simulation with GPS at the core is essentially a highly precise method for vehicle routing problems. The actual routing can be done using various multimodal approaches like Ant Colony Optimization (ACO), Maze Track Mapping, Heuristic Search, etc.**

***Keywords:* Multi-objective, Optimisation, Routing Problems, Fast route computation, Path Finding, Traffic analysis.**

I. INTRODUCTION

The rising road traffic has resulted in a lot of road congestions in recent times. Metropolitan cities are clogged with herds of cars honking and finding ways out of the congestion. But this congestion can only be solved if there is a centralized method of monitoring all the cars or vehicles moving around. Currently there is no centralized system that provides such facilities. To provide a centralized monitoring of vehicles all the vehicles in network are supposed to have Global Positioning System or any kind of location monitoring devices. Using such devices the overall volume of vehicles can be mapped and thus the overall traffic can be mapped. Transport agencies are in process of developing the Traffic Management and Control Centers (TMCCs) which map road traffic by using urban road traffic control systems whose main goals are automatic incident detection, real-time traffic information provision, and the coordination of traffic control and surveillance systems. In theory, most of the urban traffic control systems is short-term traffic forecasting.

The before mentioned problem can be addressed through different methodologies, some of the most popular methods in the past decade are Bayesian inference, time series models, and neural networks. The problem can be studied also with traffic assignment models whose aim is to predict link flows and travel times on a traffic network.

This project aims to build this mechanism and put it to test with the most extreme physical conditions on the road which could be emulated, and to resolve traffic on a timely basis. The system will be build using randomization techniques for traffic. Simulation is based on the process of modelling a real phenomenon with a set of mathematical formulas. Methodologies used for the construction of traffic simulations need to be examined in the context of real world big traffic data. This data can be used to create models for vehicle arrivals, turning behaviour, and traffic flow.

II. RELATED WORK

A. Hierarchical Time-Dependent Shortest path Algorithm for Dynamic Traffic Assignment System [1]

The briefest way issue emerges regularly in transportation field regarding the vehicle-route and the system movement task applications. Though the connections between the two application zones are self-evident, the majority of the arrangement techniques for various kinds of issues are produced independently from each other and, there is not really any most limited way calculation adaptable and sufficiently proficient for both task and route purposes, although in this framework The general outcomes imitate the truth intently.

It uses a recursive time slot hierarchy so the computation are heavy on spaceWeight assignment changes dynamically, so if threading is used most of the threads are kept in starvation to avoid early reads.

B. An Approach to Dynamical Classification of Daily Traffic Patterns [2]

In these Traffic Management and Control Centers (TMCCs) by utilizing urban traffic control frameworks whose fundamental objectives are programmed occurrence identification, constant traffic data arrangement, and the Coordination of traffic control and reconnaissance frameworks. It is heuristic based on ground truth system and it is highly effective and fast.

III. PROPOSED SYSTEM

The proposed architecture consist of (i) Global Positioning System (GPS), (ii) Road network (Digital map), (iii) On board Communication, (iv) Database of links with known traffic conditions, and (v) Packet generation unit Global Positioning System (GPS)and Road network map are tied up with each other. Global Positioning System (GPS) provides location of other cars. Road network map is there to find location of each car's position and this information is sent to Traffic Estimation. On board location unit sends location of user. Database of links with known traffic conditions consist of road constraint that road conditions are also been estimated with traffic manipulating. First, the road constraints are stored in database, and then they are sent for traffic manipulating. If those condition match, the optimal solution will be granted.

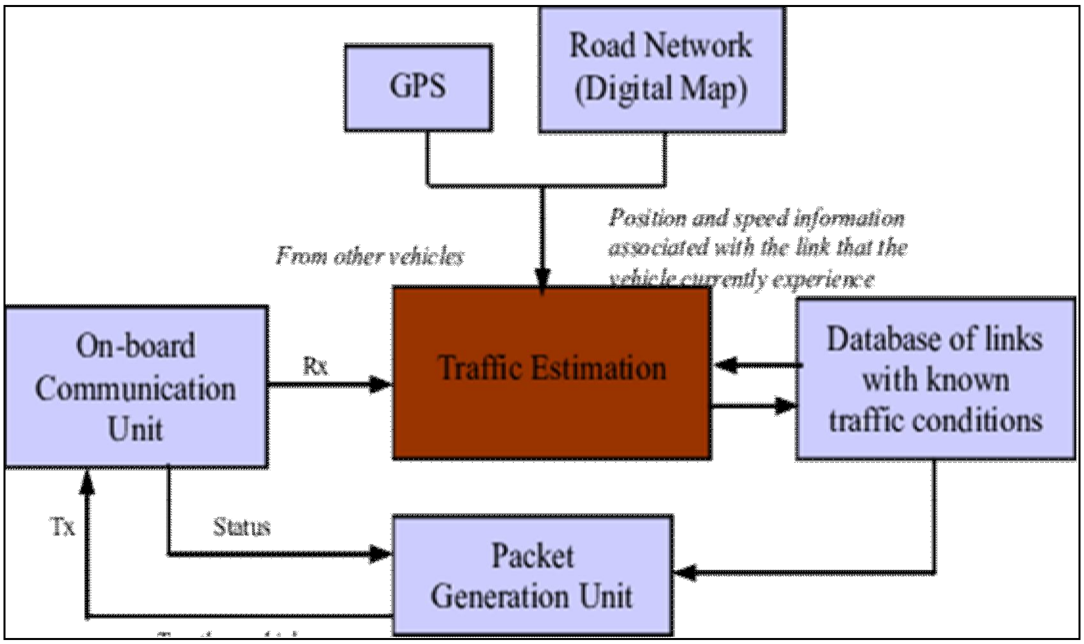


Fig.1 Architecture of System

IV. METHODOLOGY

The main idea of our knapsack based k-shortest-paths algorithm, then, is to translate the problem from one with two terminus, s and t, to a problem with only one depot. One can find track from s to t simply by finding track from s to any other acme and concatenating a shortest way from that peak to t. However, we cannot simply apply this idea directly, for several grounds:

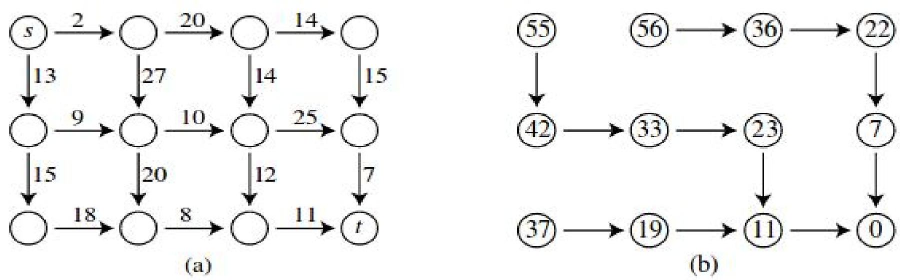
There is no obvious relation between the ordering of the path from s to other vertices and the ordering of the corresponding paths from s to t.

Each path from s to t may be represented in many ways as a path from s to some peak followed by a shortest path from that vertex to t.

Our input graphical record may not have bounded degree.

A. Preconditions

We accept all through that our information chart G has n vertices and m edges. We permit self-circles and different edges, so m might be bigger than (n/2). The length of an edge e is meant l (e). By augmentation we can characterize the length l (p) for any way in G to be the total of its edge lengths. The separation d (s, t) for a given match of vertices is the length of the most brief way beginning at s and closure at t; with the possibility of no negative cycles, this is all around characterized. Note that d (s, t) possibly unequal to d (t, s). The two endpoints of a coordinated edge e are meant tail (e) and head (e); the edge is coordinated from tail (e) to head (e).



(a) *Example digraph G with edge lengths and specified terminals;* (b) *shortest path tree T and distances to t in G.*

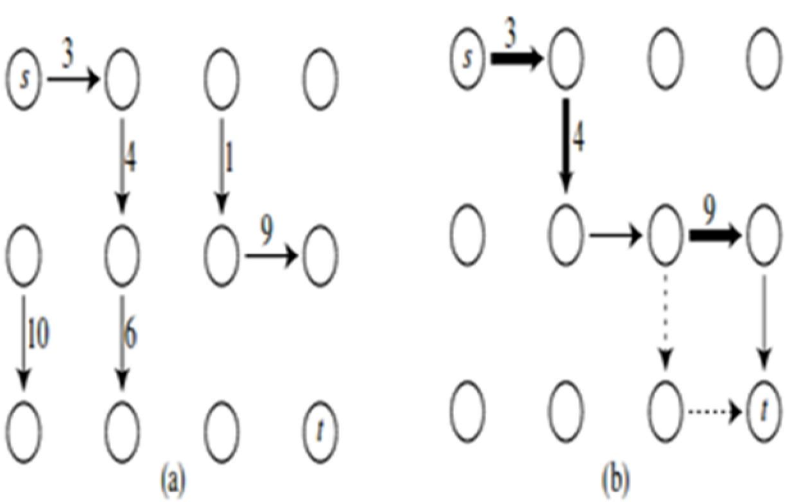
Fig.2 Traffic Route Map

B. Analysis

The I[th] shortest path in diagraph could have Ω (ni) edges, that the best time we could hope for in an exact listing of shortest ways would be O (k2n). Our time bounds are quicker than this, thus we should use an implicit illustration for the paths. However, our illustration isn't a significant obstacle to use of our algorithm: we are able to list the sides of any path we have a tendency to output in time proportional to the amount of edges, and straightforward properties are accessible in constant time. Similar implicit representations have previously been used for connected issues like the k minimum weight spanning trees.

Our representation is similar in spirit to those used for the k minimum weight spanning trees problem: for that problem, each successive tree differs from a previously listed tree by a swap, the insertion of one edge and removal of another edge. The implicit representation consists of a pointer to the previous tree and a description of the swap. For the shortest path problem, each successive path will turn out to differ from a previously listed path by the inclusion of a single edge not part of a shortest path tree and appropriate adjustments in the portion of the path that involves shortest path tree edges. Our implicit representation consists of a pointer to the previous path, and a description of the newly added edge. Given an edge e in G, define.

$$\delta(e) = \ell(e) + d(head(e), t) - d(tail(e), t).$$



(a) *Edges in G − T labeled by δ(e) (δ(e) = 0 for edges in T);* (b) *path p, sidetracks(p) (the heavy edges, labeled 3, 4, and 9), and prefpath(p) (differing from p in the two dashed edges; sidetracks(prefpath(p)) consists of the two edges labeled 3 and 4).*

Fig.3 Path Traversing In Knapsack

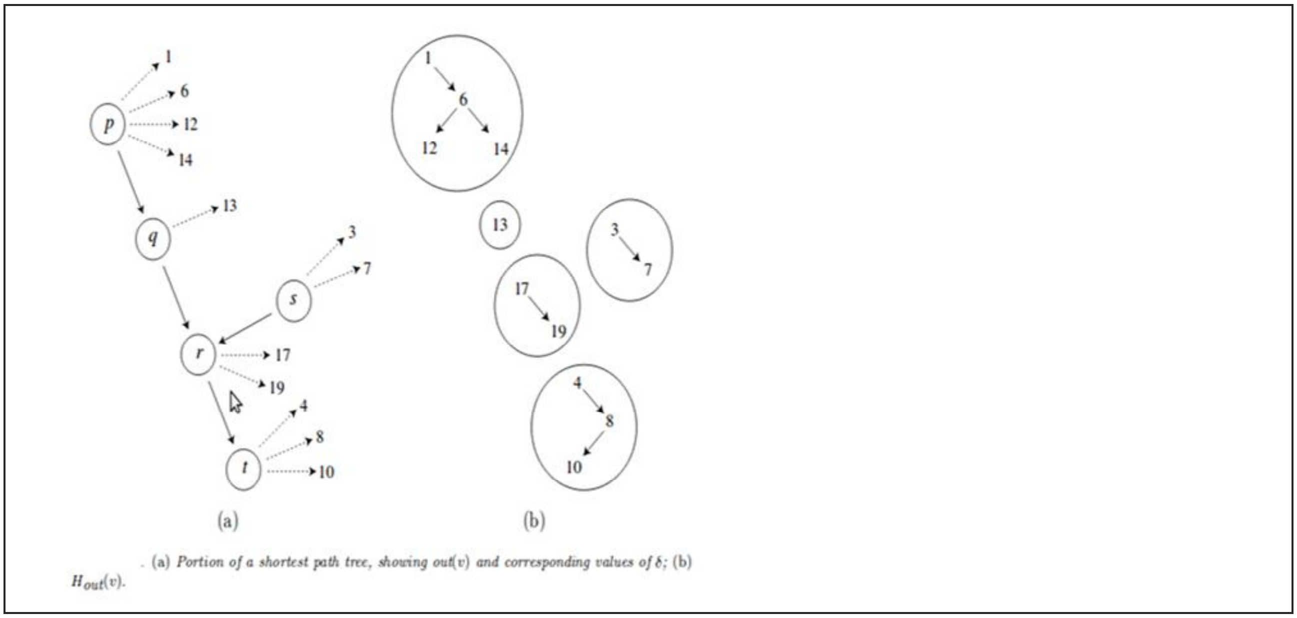


Fig.4 Multi-objective Optimisation for Broken Down Paths

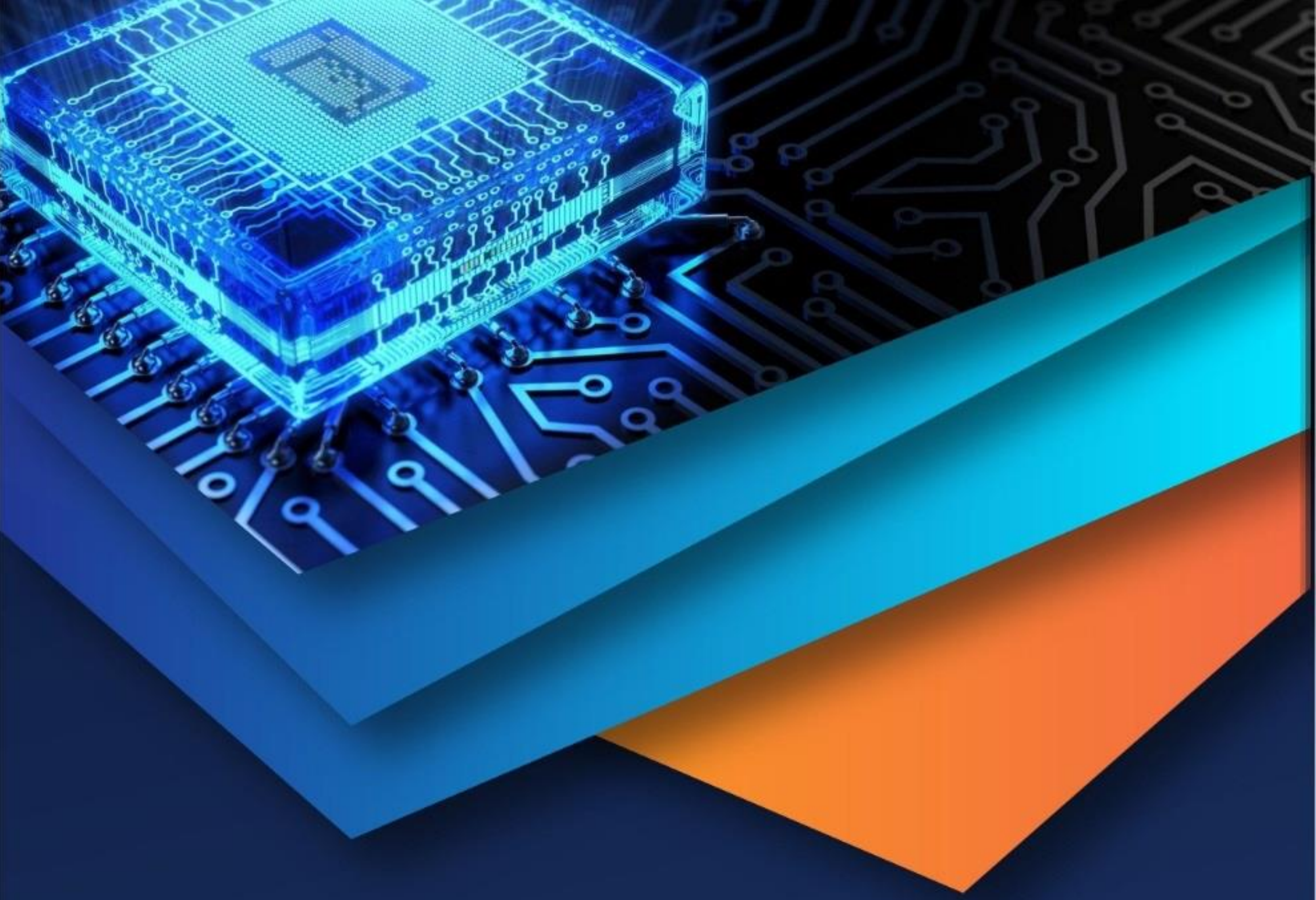
If G contains cycles, the path tree is infinite. However, since its degree is not necessarily constant, we cannot directly apply breadth first search to find its k minimum values. Instead we form a heap by replacing each node p of the path tree with an equivalent bounded-degree subtree (essentially, a heap of the edges with tails on the path from head (last-side-track (p)) to t, ordered by δ (e)). We must also take care that we do this in such a way that the portion of the path tree explored by our algorithm can be easily constructed. For each vertex v we wish to form a heap HG (v) for all edges with tails on the path from v to t, ordered by δ (e). Use heap to modify the path tree by replacing each node p with a copy of HG (head (last-side-track (p))).

V. CONCLUSION AND FUTURE WORK

This proposed system tries to address the essential phases of traffic analysis. In the future it would be possible to add a pre-processing of noisy traffic data and a pre-classification into working days and non-working days, and to analyse the performance of the entire system. Moreover, although the design of the system considers a static repository of traffic patterns, the on-line phase updates the available historical data making possible the incorporation of automatic rules to update the repository when new mobility patterns are beginning to emerge in the network by using automatic machine learning techniques.

REFERENCES

[1] Weise T, Zapf M, Chiong R, Nebro AJ. Why is optimization difficult? In: Chiong R, editor. Nature-inspired algorithms for optimisation. Studies in computational intelligence, vol. 193/2009. Springer; 2009.
[2] Glover FW, Kochenberger GA. Handbook of metaheuristics. Kluwer; 2003.
[3] W. Shih, A branch and bound method for the multi-constraint zero–one knapsack problem, Journal of the Operational Research Society 30 (1979).
[4] Pallottino, S. and M.G. Scutella, Shortest Path Algorithms in Transportation models: classical and innovative aspects. 1997, UNIVERSITA DI PISA.
[5] Cherkassky, B.V., A.V. Goldberg, and T. Radzik, Shortest Paths Algorithms: Theory and experimental evaluation. Mathematical Programming, 1996. 73(2): p. 129-174.
[6] Chabini, I., Discrete Dynamic Shortest Path Problems in Transportation Applications Complexity and Algorithms with Optimal Run Time. Transportation Research Record, 1998. 1645: p. 170-175.
[7] Kaufman, D.E. and R.L. Smith, Fastest paths in time-dependent networks for intelligent vehiclehighway systems application. IVHS Journal, 1993. 1(1): p. 1-11.
[8] P.C. Gilmore, R.E. Gomory, The theory and computation of knapsack functions, Operations Research 14 (1966) 1045–1075.
[9] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, New York, 1990.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY